Week 1 – Unix Basics (Jan 5–Jan 10, 2026)

UNIX PHILOSOPHY:
- Do one thing and do it well
- Everything is a file
- Small tools combined using pipes

Kernel vs Shell:
Kernel = Core of OS, interacts with hardware
Shell = Interface between user and kernel

CLI vs GUI:
CLI = Command based, faster, lightweight
GUI = Graphical, user-friendly

Filesystem Hierarchy:
/ (root)
/home
/etc
/bin
/usr

Terminal Usage:
Open terminal
Run commands
Use man command for help

Unix File System Concepts

Absolute Path:
/home/user/file.txt

Relative Path:
./file.txt

Hidden Files:

```
ls -a
```

Inode:
Stores metadata (size, permissions, owner)
Check inode:

```
ls -i filename
```

Basic Unix Commands

```
pwd
ls
cd foldername
mkdir newfolder
rmdir folder
rm file.txt
touch file.txt
cat file.txt
less file.txt
man ls
```

Advanced Unix Commands

```
grep "text" file.txt
find . -name "file.txt"
chmod 755 file.sh
chown user file.txt
```

Pipes:

```
ls | grep txt
```

Redirection:

```
ls > output.txt
```

Background Job:

```
sleep 10 &
jobs
```

Week 2 – Bash & Git Fundamentals (Jan 12–Jan 17, 2026)

Bash Script Example:

```
#!/bin/bash
name="Harshita"

if [ $name == "Harshita" ]; then
 echo "Welcome"
fi

for i in 1 2 3
do
 echo $i
done
```

Git Fundamentals

```
git init
git clone <repo>
git add .
git commit -m "Initial commit"
```

Git Internals

Staging Area:

```
git add file.txt
```

Check diff:

```
git diff
```

View log:

```
git log --oneline
```

Reset:

```
git reset --hard HEAD~1
```

Week 3 – Branching & JS Runtime (Jan 19–Jan 24, 2026)

Create branch:

```
git checkout -b feature
```

Merge:

```
git merge feature
```

Rebase:

```
git rebase main
```

Resolve conflict manually then:

```
git add .
git commit
```

GitHub Workflow

Push to GitHub:

```
git remote add origin <repo-url>
git push -u origin main
```

Pull Request:
Create PR on GitHub
Review and merge

JavaScript Runtime

JS Runtime:
Single-threaded
Uses Event Loop

Blocking:
fs.readFileSync()

Non-blocking:
fs.readFile()

Week 4 – Node.js (Jan 26–Jan 31, 2026)

Node Architecture:
Built on V8 Engine
Event-driven, non-blocking

Example:

```
const http = require('http');
http.createServer((req,res)=>{
 res.end("Hello");
}).listen(3000);
```

Node Environment Setup

Install Node:
Download from nodejs.org

Using nvm:

```
nvm install 18
nvm use 18
```

```
npm init -y
```

package.json structure:

```
{
 "name": "project",
 "version": "1.0.0",
 "scripts": {}
}
```

Node Core Module – fs

fs Module Example:

```
const fs = require('fs');

// Sync
const data = fs.readFileSync('file.txt','utf8');

// Async
fs.readFile('file.txt','utf8',(err,data)=>{
 console.log(data);
});
```